A Survey of Analytic Provenance

Phong H. Nguyen, Kai Xu, and B.L. William Wong

1 Introduction

Analytic provenance research tries to understand a user's reasoning process by examining their interactions with a visual analytic system. Visual analytics is the science of analytical reasoning facilitated by interactive visual interfaces [27]. The key role of visual analytics is to support analysts to derive insight from massive amounts of data and to make decision based on the derived knowledge. However, not only is the extracted knowledge important, but the analysis process that led to that knowledge and the rationale underlying the analysis are also of great significance [21, 13].

In 1996, Shneiderman already noticed the importance of studying user interactions in information visualisation by classifying *history* as one of the seven tasks in his *Task by Data Type Taxonomy* [24]. According to Shneiderman, information visualisation systems need to support users to review previous actions and correct mistakes because the information exploration process is typically long and complex. Since then, there has been more research on exploration history and analytic provenance in visualisation and related fields. In May 2011, the first workshop dedicated to analytic provenance was held in *CHI 2011* conference to develop a research agenda to better study analytic provenance and a call to action for further research. In that workshop, the following definition of analytic provenance was proposed, "the area of research that focuses on understanding a user's reasoning process through the study of their interactions with a visualisation is called Analytic Provenance" [21, p.33]. Besides understanding the user's reasoning process, many benefits can also be gained from analytic provenance such as recalling the analysis process, reusing performed analyses, supporting evidence in constructing the reasoning process, and facilitating collaboration between colleagues including dissemination, discussion and presentation (Section 4).

Typically, an *analytic provenance aware* system consists of three stages: capturing the provenance of the analysis process, visualising the captured information, and utilising the visualised provenance. As a result, we characterise the literature of analytic provenance by these stages.

2 Capturing Analytic Provenance

The first step in capturing analytic provenance is to decide what kind of information needs to be captured. Does the system capture low-level user interactions, or high-level user intentions, or both of them? The decision may depend upon how the system subsequently uses the captured information.

Based on an empirical study, Gotz and Zhou [11] characterise visual analytic activities at multiple levels of granularity according to the semantic richness of these activities: the top-level *tasks* (high-level analytic goals), the high-level *sub-tasks* (more concrete sub-goals to fulfil the goal), the low-level *actions* (detailed analytic steps to achieve the sub-goal such as filtering or sorting data) and the bottom-level *events* (the actual interactions need to perform such as mouse-clicks or keystrokes). Figure 1 illustrates the model and an example scenario.

Following this characterisation, a system can capture the information corresponding to one or many tiers. We describe the existing work of capturing provenance of each tier below.

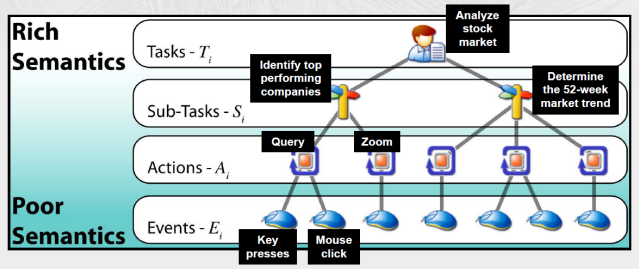


Figure 1: Illustration of the Gotz and Zhou model [11] with an example in business environment. One possible analysis task is to analyse the stock market to invest. Two proposed sub-tasks are identifying the top performing companies and finding the trend of the market this year. To accomplish the first sub-task, the analyst queries top 50 highest profit companies, which requires clicking on the query button and typing '50'. All these task, sub-tasks, actions and events are represented in the model.

2.1 Capturing Bottom-Level Events

Glass Box [4] can record a great deal of low-level information including keyboard/mouse events, window events, file open and save events, copy/paste events, and so on. Its objective is to capture, archive and retrieve intelligence analysis activities.

There is not much research on capturing events because it is relatively easy and limitedly useful. Simply capturing these events alone does not provide sufficient information to understand their purpose and rationale. For example, we know that a *mouse click* is captured; however, what the purpose of that click was (e.g., to sort the data?), and why the user performed that click (e.g., to find an interesting pattern from the data?) are unknown. Commonly, when analysing data with a visualisation, an analyst needs to perform many operations with trials and errors to find the answer to the problem. In that case, a series of poor-semantic and bottom-level events makes it more difficult for the analyst to recall what has been done. Therefore, more meaningful interactions also need to be captured.

2.2 Capturing Low-Level Actions

2.2.1 Taxonomy of Actions

Actions in the action tier are both semantic and generic across different visual analytic systems, thus are commonly used as the semantic building blocks for the provenance of derived insight [12, 25, 11]. Gotz and Zhou [11] provide a taxonomy of actions that contains the most common analytic operations in many visual analytics systems they observed. The taxonomy classifies actions, based on their intention, into three groups: *exploration* actions (e.g., *filter* the data according to a condition), *insight* actions (e.g., *bookmark* the current visualisation), and *meta* actions (e.g., *undo/redo* a performed action).

Considering an example to distinguish between capturing meaningful actions and less-meaningful events. The action *zoom-in* has an intention of increasing the details of the display area, and thus can be used either to reduce the focus area of a map or to refine the

display time range of a timeline, in a semantically equivalent manner. Moreover, this action could be accomplished by three different events: scrolling the mouse, pressing the combination of Ctrl and plus (+), and tapping on a smart device. Recording only the event; for instance, scrolling the mouse, is not sufficient to know whether its action is zoom or just a common scroll in a text editor.

2.2.2 Automatically Capturing Low-Level Actions

During the course of analysis with a visualisation, all user interactions can be systematically recorded. The visual exploration process can be modelled using *graph* metaphor. Nodes in the graph represent *states* of the application and edges represent *actions* that transform one state into the other state. Considering an example of *bar-chart* visualisation, states are all the necessary information allowing to reconstruct the captured chart such as the *dataset* and the *colour map*; while an action could be *sorting data*. The system can support *undo* to revisit to a previous state; and if a new action is performed at that state, a new branch will be created to store that new line of inquiry.

Basically, there are two prime strategies to automatically capture the exploration process. One is capturing the initial state and all the performed actions so that they can be rerun to achieve the desired state [16]. Second is simply capturing all visualisation states after each action [1]. The former strategy suffers from potential long running time if the number of actions need to executed is high; while the latter is memory-expensive if a state contain too much information. The later is easier to implement; whereas, the former allows re-applying the analysis process with a different dataset.

2.3 Capturing High-Level Sub-Tasks or User Intentions

Typically, high-level sub-tasks can be either inferred from captured low-level actions or directly recorded by users.

2.3.1 Deriving from Captured Actions

When analysts interact with a visual analytics tool, their plans and methods to analyse data could partially be reflected through their interactions with the application.

Manual Derivation Dou et al. [7] conduct a quantitative study to measure how much of a user's reasoning process can be recovered from only the captured user actions. Reasoning results decoded from the interaction logs are compared with the ground-truth reasoning from analysts' interviews; and the results show that 79 per cent of the findings, 60 per cent of the methods and 60 per cent of the strategies could be extracted from manually analysing the interaction logs. This post-analysis approach is domain-specific because ad hoc tools need to be designed to effectively discover some well-known strategies in a particular domain, detecting suspicious activities in wire transactions. Even though reasoning processes are discovered, the interaction analyses occur after the data analyses and thus cannot support analysts in real-time.

Automatic Derivation Gotz and Zhou use heuristics to automatically infer a sub-task from a series of actions [11]. One heuristic suggests that a user solves a sub-task by completing a combination of several exploration actions followed by an insight action. For example, the analyst explores the data by selecting bar-chart as a visualisation technique (*change-metaphor* action), sorting the data according to some indicator (*sort* action), and then annotating (*annotate* action) on the highest column of the chart. The heuristic considers "annotate", the insight action, as a signal of deriving insight, or solving a sub-task; and represents that sub-task as a trail of three actions "change-metaphor - sort - annotate". However, if the analyst does not annotate or bookmark visualisations, the heuristic cannot derive any sub-tasks.

Automatic derivation provides real-time support for users to quickly understand the analysis process. However, because heuristic approach could lead to a misleading user intention, it should only assist analysts and allow them to correct the derived intention.

2.3.2 Directly Capturing using Annotations

Instead of inferring user intentions from low level actions, analysts can manually capture the insight by annotating on the visualisations of interest. Sense.us [14], a web site supporting asynchronous collaboration, allows users to annotate on visualisations, and use these annotated visualisations in discussion. GeoTime [9], a geotemporal event visualisation tool, supports embedding hypertext linked visualisations and visual annotations in an analysis story. Annotation can provide more information than simple text and graphics attachment. *Data-aware* annotation detects the subset of data belonging to the annotated area so that the data of interest remains unchanged when new visualisation metaphors are applied for further investigation [5]. Another benefit of data-aware annotation is that statistical values could be automatically generated to add more information such as the mean and the extreme values of selected data items [3].

Manual annotation provides high-fidelity; however, users often only take notes of the final state of a visualisation [11]. Therefore, intuitive annotation mechanism needs to be designed to encourage users to take notes.

2.4 Capturing Top-Level Tasks

Top-level tasks are highly domain-specific; therefore, it's virtually impossible to automatically derive them. The user needs to explicitly write down what the task is before solving the problem. Further enhancement has been made to allow users to document their reasoning processes; for example, recording found interesting patterns about the data, describing their causal relationships, and building a hypothesis based on these found artefacts [25, 22]. This mental model needs to be documented directly onto the same system for effective reasoning rather than keeping tacitly or recording it into an external application such as Microsoft Word (see [25] for explanation).

3 Visualising the Captured Information

Typically, events are not visualised because they do not carry much information and the number of events is high. Actions and states (the visual results of the actions) are commonly visualised together to depict the *analysis process*. Sub-tasks and tasks are illustrated in the graphical *reasoning process*.

3.1 Visualising the Analysis Process

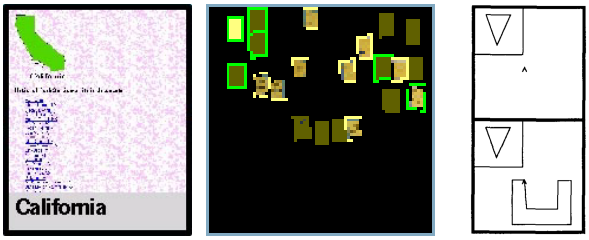
Methods of visualising individual actions and states will be discussed, and followed by methods of chaining them together to visualise the entire analysis process.

3.1.1 Visual Representation of States

Conventionally, the space for rendering a state is limited because a small portion of the display area is reserved for provenance information; whereas, the much larger portion is used for data exploration [25, 11, 16]. Therefore, a small-scale visualisation of the captured state is popular [15] (Figure 2(a)). To recall the affect of the performed action, the miniature can highlight the difference from the previous state [17] (Figure 2(b)), or combine both the former and the latter visualisations corresponding to that action [18] (Figure 2(c)).

3.1.2 Visual Representation of Actions

Typically, a system supports a certain number of actions; and thus allows using icons to visually distinguish different kinds of actions besides texts [11] (Figure 3(a)). Actions are also commonly represented as edges in a graph to connect two states. Therefore, graph



(a) The exact miniature of a web-site[15].

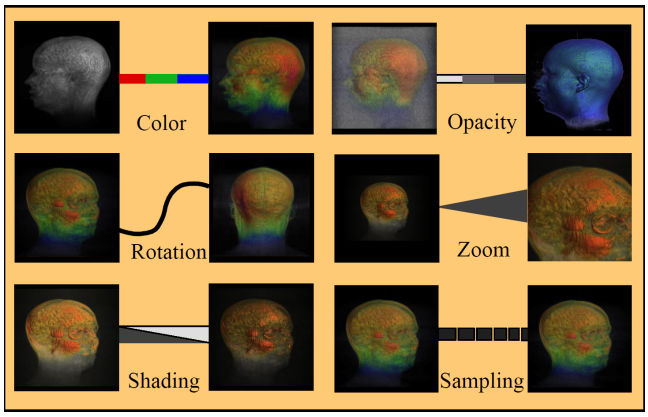
(b) Each rectangle represents a post-it note. Modified post-it notes are highlighted with green colour [17].

(c) In a graphical editor, combining the state before the action (above) and the state after the action (bottom) [18].

Figure 2: Examples of visual representations of states.

edges can be stylised to reflect the characteristics of the represented actions [19] (Figure 3(b)).

(a) Using icons to represent actions including query, filter, change-view and inspect [11].



(b) Using stylish edges to represent actions including changing colour map, rotating, shading, changing opacity, zooming and sampling [19].

Figure 3: Examples of using visual representations of different types of actions.

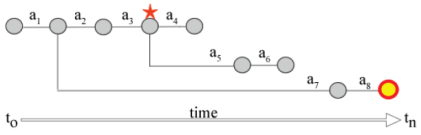
3.1.3 Layout of Actions and States

Typically, the system begins with an initial state (node). When the user performs an action, a new node is created for the current state, and a new edge is added to connect the previous node with the current node. Gradually, a string of nodes and edges is built in the chronological order. The system can support revisiting to the previous state. If a new action is performed at that state, a new branch will be forked to store forthcoming actions. Therefore, the analysis process has the layout of a *direct acyclic graph*, or a *tree* if revisited links are not explicitly visualised.

To not distract analysts from the primary data exploration and save space, several techniques have been proposed to reduce the display area of the provenance graph: organising trees in the right horizontal-vertical layout [25], displaying only nodes of the active branch that led to the selected visualisation [17], allowing graph nodes be expandable/collapsible on demand [1], supporting zoomable and pan-able interface [8], and applying distortion techniques to focus on more relevant states [20].

The order of actions can be interpreted through the direction of edges in the provenance graph. Moreover, exact time gap between actions can also be measured and visually encoded into the visualisation. VisTrails [1] colour-codes the background of visualisation nodes according to when they are created (Figure 4(a)); and Aruvi [25] uses the length of edges to represent the distance in terms of time between two states (Figure 4(b)). This time indication can be updated only when a new node is added, or continuously to reflect the fact that time is always flying. In the latter case, endlessly, background nodes will become lighter and edge lengths will become shorter. This *time-travel* interface is implemented in Visage [6].

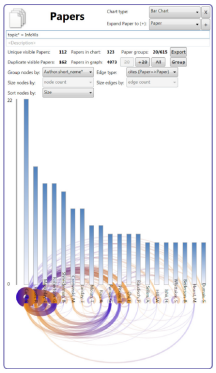
(a) The darker node refers to the later visualisations [1].



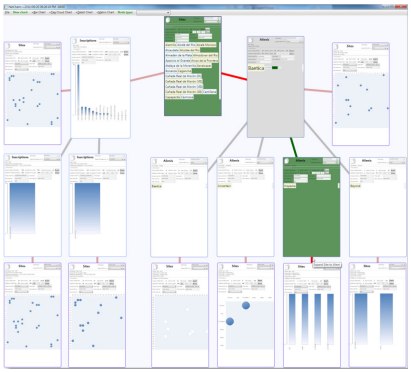
(b) The edge length represents time distance between two connecting nodes [25].

Figure 4: Examples of encoding time into provenance visualisation.

The provenance space can be unified with the data exploration space, where each node of the provenance graph represents a fully interactive visualisation. Zoom-able and pan-able interface needs to be supported to allow users either to focus on the current visualisation (Figure 5(a)) or to observe the entire context of the analysis process (Figure 5(b)).



(a) Focus view on the active visualisation.



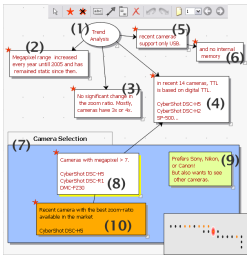
(b) Context view to see the overview of what has been done.

Figure 5: An example of integration of the provenance space into the data exploration space [8].

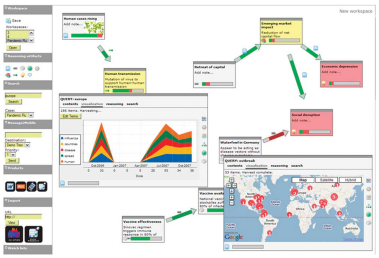
3.2 Visualising the Reasoning Process

The Aruvi system [25] allows analysts to freely compose the reasoning process by using a graphical editor. Users can take note in rectangles or ellipses, and use arrows to connect them. Conventionally, nodes can be referred as *evidence*, *assumptions* or *hypotheses*, and arrows can be referred as *causal relationships*. Nodes in the editor can be linked to the captured visualisations to help explain its reasoning, and these nodes are marked with a star to indicate the existence of the linked visualisations. Instead of using a graphical editor and an implicit convention to map drawing shapes with reasoning artefacts, Scalable Reasoning System [22] provides a more formal method to document the reasoning process. A captured visualisation can be dropped to the reasoning space to create a node.

The node shows the miniature of the captured visualisation and can be tagged as an *evidence* artefact. An evidence can be converted to a *casual relationship* and its rectangular shape will become an edge. An *assumption* is a free note and can be upgraded as a *hypothesis* when it is supported by an evidence. Figure 6 shows those two examples of reasoning process visualisation.



(a) Using graphical editor to freely construct the mental model [25].



(b) A formal reasoning diagram with different types of artefacts: evidence, casual relationship, assumption and hypothesis [22].

Figure 6: Examples of visualising the reasoning process.

4 Utilising the Visualised Provenance

The visualised provenance can be exploited to support the analytical reasoning process and collaboration.

4.1 Supporting the Analytical Reasoning Process

4.1.1 Recalling the Analysis Process

Provenance visualisation provides a visual overview of the analysis process. Therefore, it helps the analyst recall what has been done, and potentially reminds what is missing and what should be done. Provenance visualisation should not be a static picture of the past. In contrast, it should allow the analyst to freely navigate back to the desired state [6, 17, 1]. A state can be easily selected through the interface or with the help of *search and filter* mechanism when there are too many recorded states. During the analysis process, users can take notes and tag keywords on visualisations; and all these metadata are subjects to search [26]. Moreover, VisTrails supports *query-by-example* to find related visualisations [23]. Past states can be filtered to display periodically [17] or based on a particular metadata such as authors [17] and time [26].

4.1.2 Reusing the Performed Analyses

When reviewing the provenance, analysts can insert missing actions, remove undesired actions, and reapply past actions to a new dataset [6]. The past actions can also be modified directly by changing the command statements [16], the command parameters [10], and the changing effects can be propagated along the history trail [19].

Typically, in scientific visualisation, a visualisation is considered as a rendering result of many understandable parameters. Therefore, it is feasible to compare visualisations by measuring these sets of parameter values. VisTrails, [1], a scientific visualisation workflow system, allows comparing and merging two visualisations into a new one. More specifically, *set* operations including *intersection*, *union* and *difference* can be employed to build the parameter set of the generated visualisation. In GraphTrail [8], a information visualisation tool, it is still possible to merge visualisations in terms of data. The active dataset of each visualisation is mapped with a *SQL statement*; thus, performing a SQL *union* statement will result a new visualisation with the combined data of interest.

4.1.3 Supporting Evidence in Constructing the Reasoning Process

As discussed in Section 3.2, the reasoning process can be graphically documented inside the system. By capturing analytic provenance, we can attach the recorded visualisation to the reasoning evidence to support that artefact [25]. Not only the visualisation but also could all the steps that the analyst performed to generate that visualisation be helpful.

4.2 Supporting Collaboration

4.2.1 Dissemination and Discussion

Visualisations can be annotated, captured and attached into the discussion forum to help peers understand the findings of author easier [14]. Captured provenance can also be embedded into a formal analysis story to visually convey idea [9]. The embedded provenance should be interactive so that audience can examine and verify what the author wrote [22]. In asynchronous collaboration, each individual colleague can capture insight and submit them to a central repository. As a result, all peers can exploit other findings and facilitate the solving problem process [2].

4.2.2 Presentation

Analytic provenance can also be exported for presentation purpose with various published formats. Outpost [17], a tangible interface for collaborative web site design, provides a *print version* of annotated visualisations as a report. VisTrails [1], a scientific workflow and provenance management system, supports embedding the visualisation process into a paper through *Latex* format. Image Graphs [19], a volume visualisation system, builds an *animation* from selected key visualisations.

References

[1] L. Bavoil, S. Callahan, P. Crossno, J. Freire, C. Scheidegger, C. Silva, and H. Vo. VisTrails: Enabling Interactive Multiple-View Visualizations. In *IEEE Symposium on Information Visualization*, pages 135–142. IEEE, 2005.

[2] Y. Chen, J. Alsakran, S. Barlowe, J. Yang, and Y. Zhao. Supporting effective common ground construction in Asynchronous Collaborative Visual Analytics. In *IEEE Symposium on Visual Analytics Science And Technology*, pages 101–110. IEEE, Oct. 2011.

[3] Y. Chen, S. Barlowe, and J. Yang. Click2Annotate: Automated Insight Externalization with rich semantics. In *IEEE Symposium on Visual Analytics Science And Technology*, pages 155–162. IEEE, Oct. 2010.

[4] P. Cowley, J. Haack, R. Littlefield, and E. Hampson. Glass box: capturing, archiving, and retrieving workstation activities. In *ACM Workshop on Continuous Archival and Retrival of Personal Experience*, pages 13–18, New York, New York, USA, Oct. 2006. ACM Press.

[5] I. Denisovich. Software Support for Annotation of Visualized Data Using Hand-Drawn Marks. In *Conference on Information Visualisation*, pages 807–813. IEEE, July 2005.

[6] M. Derthick and S. F. Roth. Enhancing data exploration with a branching history of user operations. *Knowledge-Based Systems*, 14(1-2):65–74, 2001.

[7] W. Dou, D. H. Jeong, F. Stukes, W. Ribarsky, H. R. Lipford, and R. Chang. Recovering Reasoning Processes from User Interactions. *IEEE Computer Graphics and Applications*, 29(3):52–61, May 2009.

[8] C. Dunne, N. H. Riche, B. Lee, R. A. Metoyer, and G. G. Robertson. GraphTrail: Analyzing Large Multivariate and Heterogeneous Networks while Supporting Exploration History. In *ACM Conference on Human Factors in Computing Systems*, pages 1663–1672, 2012.

[9] R. Eccles, T. Kapler, R. Harper, and W. Wright. Stories in GeoTime. *IEEE Symposium on Visual Analytics Science And Technology*, 7(1):19–26, 2007.

[10] D. Gotz, Z. When, J. Lu, P. Kissa, N. Cao, W. H. Qian, S. X. Liu, and M. X. Zhou. HARVEST: An Intelligent Visual Analytic Tool for the Masses. In *International Workshop on Intelligent Visual Interfaces*

for Text Analysis, pages 1–4, New York, New York, USA, Feb. 2010. ACM Press.

[11] D. Gotz and M. X. Zhou. Characterizing users visual analytic activity for insight provenance. *Information Visualization*, 8(1):42–55, Jan. 2009.

[12] D. P. Groth and K. Streefkerk. Provenance and annotation for visual exploration systems. *IEEE Transactions on Visualization and Computer Graphics*, 12(6):1500–1510, 2006.

[13] J. Heer, J. Mackinlay, C. Stolte, and M. Agrawala. Graphical histories for visualization: supporting analysis, communication, and evaluation. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1189–1196, 2008.

[14] J. Heer, F. B. Viégas, and M. Wattenberg. Voyagers and Voyeurs: Supporting Asynchronous Collaborative Visualization. *Communications of the ACM*, 52(1):87–97, Jan. 2009.

[15] R. R. Hightower, L. T. Ring, J. I. Helfman, B. B. Bederson, and J. D. Hollan. Graphical Multiscale Web Histories: A Study of PadPrints. In *ACM Symposium on User Interface Software and Technology*, pages 121–122, New York, New York, USA, Nov. 1998. ACM Press.

[16] N. Kadivar, V. Chen, D. Dunsmuir, E. Lee, C. Qian, J. Dill, C. Shaw, and R. Woodbury. Capturing and supporting the analysis process. In *IEEE Symposium on Visual Analytics Science And Technology*, pages 131–138. IEEE, 2009.

[17] S. R. Klemmer, M. Thomsen, E. Phelps-Goodman, R. Lee, and J. A. Landay. Where do web sites come from? Capturing and Interacting with Design History. In *ACM Conference on Human Factors in Computing Systems*, pages 1–8, New York, New York, USA, Apr. 2002. ACM Press.

[18] D. Kurlander and S. Feiner. Editable graphical histories. In *IEEE Workshop on Visual Languages*, pages 127–134. IEEE Comput. Soc. Press, 1988.

[19] K.-L. Ma. Image graphs - a novel approach to visual data exploration. In *IEEE Conference on Visualization*, pages 81–88, Oct. 1999.

[20] C. Meng, M. Yasue, A. Imamiya, and M. Xiaoyang. Visualizing histories for selective undo and redo. In *Asian Pacific Computer and Human Interaction*, pages 459–464. IEEE Comput. Soc, 1998.

[21] C. North, R. Chang, A. Endert, W. Dou, R. May, B. Pike, and G. Fink. Analytic provenance: process+interaction+insight. In *ACM Transactions on Computer-Human Interaction*, pages 33–36. ACM, May 2011.

[22] W. Pike, J. Bruce, B. Baddeley, D. Best, L. Franklin, R. May, D. Rice, R. Riensche, and K. Younkin. The Scalable Reasoning System: Lightweight visualization for distributed analytics. *Information Visualization*, 8(1):71–84, 2009.

[23] C. Scheidegger, H. Vo, D. Koop, J. Freire, and C. Silva. Querying and creating visualizations by analogy. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1560–1567, 2007.

[24] B. Shneiderman. The Eyes Have It : A Task by Data Type Taxonomy for Information Visualizations. In *IEEE Symposium on Visual Languages*, pages 336–343, 1996.

[25] Y. B. Shrinivasan and J. J. van Wijk. Supporting the Analytical Reasoning Process in Information Visualization. In *ACM Conference on Human Factors in Computing Systems*, pages 1237–1246, New York, New York, USA, Apr. 2008. ACM Press.

[26] Y. B. Shrinivasan and J. J. van Wijk. Supporting Exploration Awareness in Information Visualization. *IEEE Computer Graphics and Applications*, 29(5):34–43, Sept. 2009.

[27] J. J. Thomas and K. A. Cook. *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. National Visualization and Analytics Center, 2005.